Basi di dati II — Prova parziale — 30 marzo 2015 — Compito A

Tempo a disposizione: un'ora e quindici minuti.

Si suggerisce di scrivere prima una brutta copia, per indicare poi negli spazi le risposte e brevi giustificazioni.

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (15%)

Considerare una tabella `R` appena creata (e quindi vuota), con le seguenti ipotesi

- `R` è definita su due campi, `A` di lunghezza $a = 6$ byte e `B` di lunghezza $b = 12$ byte, senza vincoli espliciti di chiave (e quindi le operazioni si possono fare senza verifiche particolari);
- la struttura fisica utilizzata per `R` è heap, senza indici, con una memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi ne servano 2 ulteriori per la memorizzazione) e in cui si marcano come liberi gli spazi dei record eliminati, **senza riutilizzarli per successivi inserimenti** (se non dopo una **riorganizzazione** che ricompatti i blocchi);
- il sistema utilizza blocchi di dimensione $D = 2$ Kbyte (approssimabili a 2000).

In tale contesto, supporre che vengano eseguite le seguenti operazioni

1. inserimento di $N = 100.000$ ennuple
2. eliminazione di $N/2 = 50.000$ ennuple (sulla base di una condizione verificabile durante la scansione)
3. dopo la conclusione e la chiusura della scansione precedente, inserimento di altre N ennuple
4. riorganizzazione del file con ricompattazione dei blocchi

Rispondere alle domande seguenti, indicando formule e valori numerici:

Fattore di blocco f per la relazione `R`:
Numero dei blocchi occupati da `R` dopo la prima serie di inserimenti (punto 1):
Numero dei blocchi occupati da `R` dopo le eliminazioni di cui al punto 2:
Numero dei blocchi occupati da `R` dopo la seconda serie di inserimenti (punto 3):
Numero dei blocchi occupati da `R` dopo la ricompattazione (punto 4):

Basi di dati II — 30 marzo 2015 — Compito A

Domanda 2 (30%) Considerare un sistema con blocchi di dimensione $P = 8$ KByte e

(a) una base di dati con una relazione *R(A B C D E)*, in cui gli attributi hanno tutti la stessa dimensione $a = 40$ Byte, . Si supponga che la relazione contenga $N = 20.000.000$ ennuple

(b) una base di dati con una coppia di relazioni R_1*(A B C)* e R_2*(A D E)* ottenute per proiezione dalla relazione R di cui al punto (a)

e le operazioni seguenti:

o_{1a} `SELECT * FROM R ORDER BY A` (su (a))
o_{1b} `SELECT * FROM R1 JOIN R2 ON R1.A = R2.A ORDER BY R1.A` (su (b))
o_{2a} `SELECT A, B, C FROM R` (su (a))
o_{2b} `SELECT A, B, C FROM R1` (su (b))

Indicare i costi relativi all'esecuzione delle varie operazioni, supponendo che, in tutti i casi, l'ordinamento possa essere realizzato con due passate di un merge sort a più vie (e che il join possa essere effettuato con un merge-scan senza materializzare il risultato dei due ordinamenti).

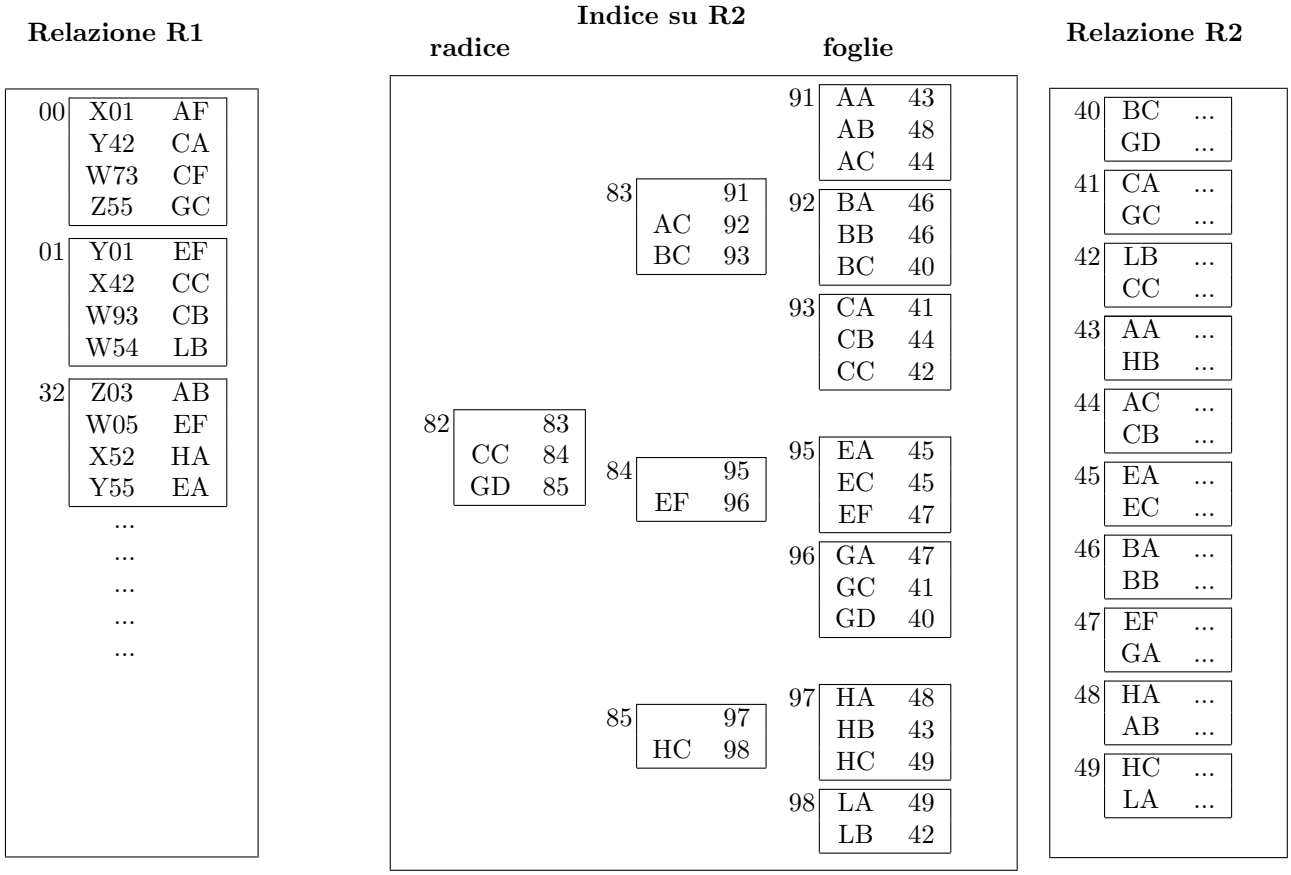
blocchi di R: (indicare nel seguito con B_a)	blocchi di R_1 (indicare nel seguito con B_{b1}) blocchi di R_2 (indicare nel seguito con B_{b2})
costo di o_{1a} (indicare nel seguito con c_{1a})	costo di o_{1b} (indicare nel seguito con c_{1b})
costo di o_{2a} (indicare nel seguito con c_{2a})	costo di o_{2b} (indicare nel seguito con c_{2b})

Le due basi di dati (a) e (b) possono essere considerate due alternative di memorizzazione e le operazioni o_{1a} e o_{1b} sono fra loro equivalenti, così come lo sono o_{2a} e o_{2b}. Quindi è interessante valutare quale delle due alternative sia conveniente.

Supponendo che l'operazione o_1 venga eseguita con frequenza f e l'operazione o_2 con frequenza $k \times f$, si intuisce che (essendo paragonabili, pur se diversi, i costi delle due operazioni) una delle alternative risulta conveniente per k molto maggiore di 1 e l'altra per k molto minore di 1. Verificare questa affermazione con riferimento a $k = 10$ e $k = 0,1$.

Basi di dati II — 30 marzo 2015 — Compito A

Domanda 3 (30%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

Domanda 4 (20%)
Si consideri un B-tree con nodi intermedi che contengono quattro chiavi e cinque puntatori e foglie con quattro chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 41, 57, 11, 32, 20, 27, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Mostrare poi l'albero dopo l'eliminazione della chiave 20 dall'ultimo albero ottenuto in risposta alla domanda precedente.

Basi di dati II — Prova parziale — 30 marzo 2015 — Compito B

Tempo a disposizione: un'ora e quindici minuti.

Si suggerisce di scrivere prima una brutta copia, per indicare poi negli spazi le risposte e brevi giustificazioni.

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (15%)

Considerare una tabella `T` appena creata (e quindi vuota), con le seguenti ipotesi

- `T` è definita su due campi, `A` di lunghezza $a = 16$ byte e `B` di lunghezza $b = 22$ byte, senza vincoli espliciti di chiave (e quindi le operazioni si possono fare senza verifiche particolari);
- la struttura fisica utilizzata per `T` è heap, senza indici, con una memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi ne servano 2 ulteriori per la memorizzazione) e in cui si marcano come liberi gli spazi dei record eliminati, **senza riutilizzarli per successivi inserimenti** (se non dopo una **riorganizzazione** che ricompatti i blocchi);
- il sistema utilizza blocchi di dimensione $D = 4$ Kbyte (approssimabili a 4000).

In tale contesto, supporre che vengano eseguite le seguenti operazioni

1. inserimento di $L = 100.000$ ennuple
2. eliminazione di $L/2 = 50.000$ ennuple (sulla base di una condizione verificabile durante la scansione)
3. dopo la conclusione e la chiusura della scansione precedente, inserimento di altre L ennuple
4. riorganizzazione del file con ricompattazione dei blocchi

Rispondere alle domande seguenti, indicando formule e valori numerici:

Fattore di blocco f per la relazione `T`:
Numero dei blocchi occupati da `T` dopo la prima serie di inserimenti (punto 1):
Numero dei blocchi occupati da `T` dopo le eliminazioni di cui al punto 2:
Numero dei blocchi occupati da `T` dopo la seconda serie di inserimenti (punto 3):
Numero dei blocchi occupati da `T` dopo la ricompattazione (punto 4):

Basi di dati II — 30 marzo 2015 — Compito B

Domanda 2 (30%) Considerare un sistema con blocchi di dimensione $P = 4$ KByte e

(a) una base di dati con una relazione *R(A B C D E)*, in cui gli attributi hanno tutti la stessa dimensione $a = 20$ Byte, . Si supponga che la relazione contenga $L = 20.000.000$ ennuple

(b) una base di dati con una coppia di relazioni *R_1(A B C)* e *R_2(A D E)* ottenute per proiezione dalla relazione R di cui al punto (a)

e le operazioni seguenti:

o_{1a} `SELECT A, B, C FROM R` (su (a))
o_{1b} `SELECT A, B, C FROM R1` (su (b))
o_{2a} `SELECT * FROM R ORDER BY A` (su (a))
o_{2b} `SELECT * FROM R1 JOIN R2 ON R1.A = R2.A ORDER BY R1.A` (su (b))

Indicare i costi relativi all'esecuzione delle varie operazioni, supponendo che, in tutti i casi, l'ordinamento possa essere realizzato con due passate di un merge sort a più vie (e che il join possa essere effettuato con un merge-scan senza materializzare il risultato dei due ordinamenti).

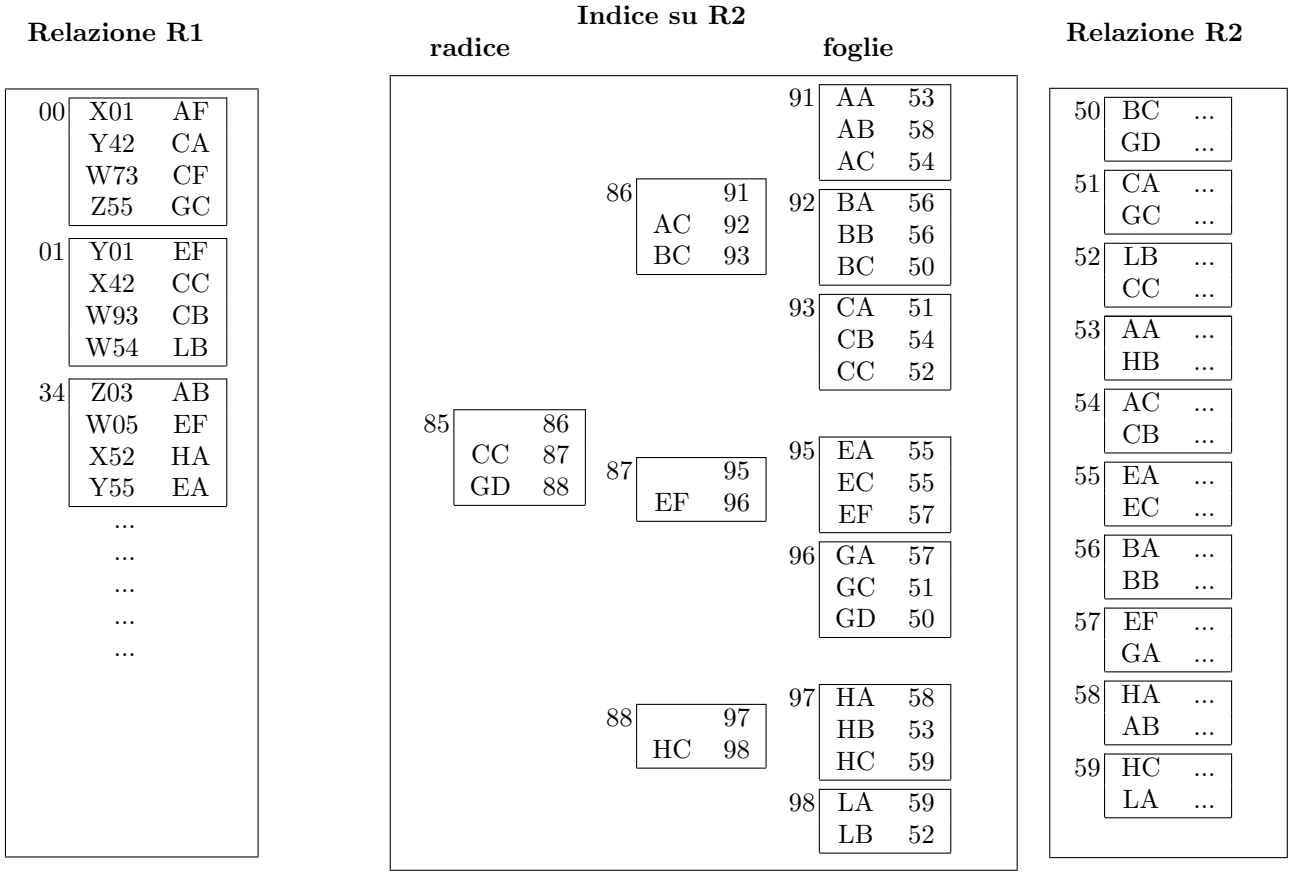
blocchi di R: (indicare nel seguito con B_a)	blocchi di R_1 (indicare nel seguito con B_{b1}) blocchi di R_2 (indicare nel seguito con B_{b2})
costo di o_{1a} (indicare nel seguito con c_{1a})	costo di o_{1b} (indicare nel seguito con c_{1b})
costo di o_{2a} (indicare nel seguito con c_{2a})	costo di o_{2b} (indicare nel seguito con c_{2b})

Le due basi di dati (a) e (b) possono essere considerate due alternative di memorizzazione e le operazioni o_{1a} e o_{1b} sono fra loro equivalenti, così come lo sono o_{2a} e o_{2b}. Quindi è interessante valutare quale delle due alternative sia conveniente.

Supponendo che l'operazione o_1 venga eseguita con frequenza f e l'operazione o_2 con frequenza $k \times f$, si intuisce che (essendo paragonabili, pur se diversi, i costi delle due operazioni) una delle alternative risulta conveniente per k molto maggiore di 1 e l'altra per k molto minore di 1. Verificare questa affermazione con riferimento a $k = 10$ e $k = 0,1$.

Basi di dati II — 30 marzo 2015 — Compito B

Domanda 3 (30%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

Domanda 4 (20%)
Si consideri un B-tree con nodi intermedi che contengono quattro chiavi e cinque puntatori e foglie con quattro chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 43, 51, 10, 32, 21, 25, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Mostrare poi l'albero dopo l'eliminazione della chiave 21 dall'ultimo albero ottenuto in risposta alla domanda precedente.

Basi di dati II — Prova parziale — 30 marzo 2015 — Compito C

Tempo a disposizione: un'ora e quindici minuti.

Si suggerisce di scrivere prima una brutta copia, per indicare poi negli spazi le risposte e brevi giustificazioni.

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (15%)

Considerare una tabella `T` appena creata (e quindi vuota), con le seguenti ipotesi

- `T` è definita su due campi, `A` di lunghezza $a = 8$ byte e `B` di lunghezza $b = 10$ byte, senza vincoli espliciti di chiave (e quindi le operazioni si possono fare senza verifiche particolari);
- la struttura fisica utilizzata per `T` è heap, senza indici, con una memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi ne servano 2 ulteriori per la memorizzazione) e in cui si marcano come liberi gli spazi dei record eliminati, **senza riutilizzarli per successivi inserimenti** (se non dopo una **riorganizzazione** che ricompatti i blocchi);
- il sistema utilizza blocchi di dimensione $D = 2$ Kbyte (approssimabili a 2000).

In tale contesto, supporre che vengano eseguite le seguenti operazioni

1. inserimento di $N = 100.000$ ennuple
2. eliminazione di $N/2 = 50.000$ ennuple (sulla base di una condizione verificabile durante la scansione)
3. dopo la conclusione e la chiusura della scansione precedente, inserimento di altre N ennuple
4. riorganizzazione del file con ricompattazione dei blocchi

Rispondere alle domande seguenti, indicando formule e valori numerici:

Fattore di blocco f per la relazione `T`:
Numero dei blocchi occupati da `T` dopo la prima serie di inserimenti (punto 1):
Numero dei blocchi occupati da `T` dopo le eliminazioni di cui al punto 2:
Numero dei blocchi occupati da `T` dopo la seconda serie di inserimenti (punto 3):
Numero dei blocchi occupati da `T` dopo la ricompattazione (punto 4):

Basi di dati II — 30 marzo 2015 — Compito C

Domanda 2 (30%) Considerare un sistema con blocchi di dimensione $P = 16$ KByte e

(a) una base di dati con una relazione *R(A B C D E)*, in cui gli attributi hanno tutti la stessa dimensione $a = 40$ Byte, . Si supponga che la relazione contenga $N = 40.000.000$ ennuple

(b) una base di dati con una coppia di relazioni R_1*(A B C)* e R_2*(A D E)* ottenute per proiezione dalla relazione R di cui al punto (a)

e le operazioni seguenti:

o_{1a} `SELECT * FROM R ORDER BY A` (su (a))
o_{1b} `SELECT * FROM R1 JOIN R2 ON R1.A = R2.A ORDER BY R1.A` (su (b))
o_{2a} `SELECT A, B, C FROM R` (su (a))
o_{2b} `SELECT A, B, C FROM R1` (su (b))

Indicare i costi relativi all'esecuzione delle varie operazioni, supponendo che, in tutti i casi, l'ordinamento possa essere realizzato con due passate di un merge sort a più vie (e che il join possa essere effettuato con un merge-scan senza materializzare il risultato dei due ordinamenti).

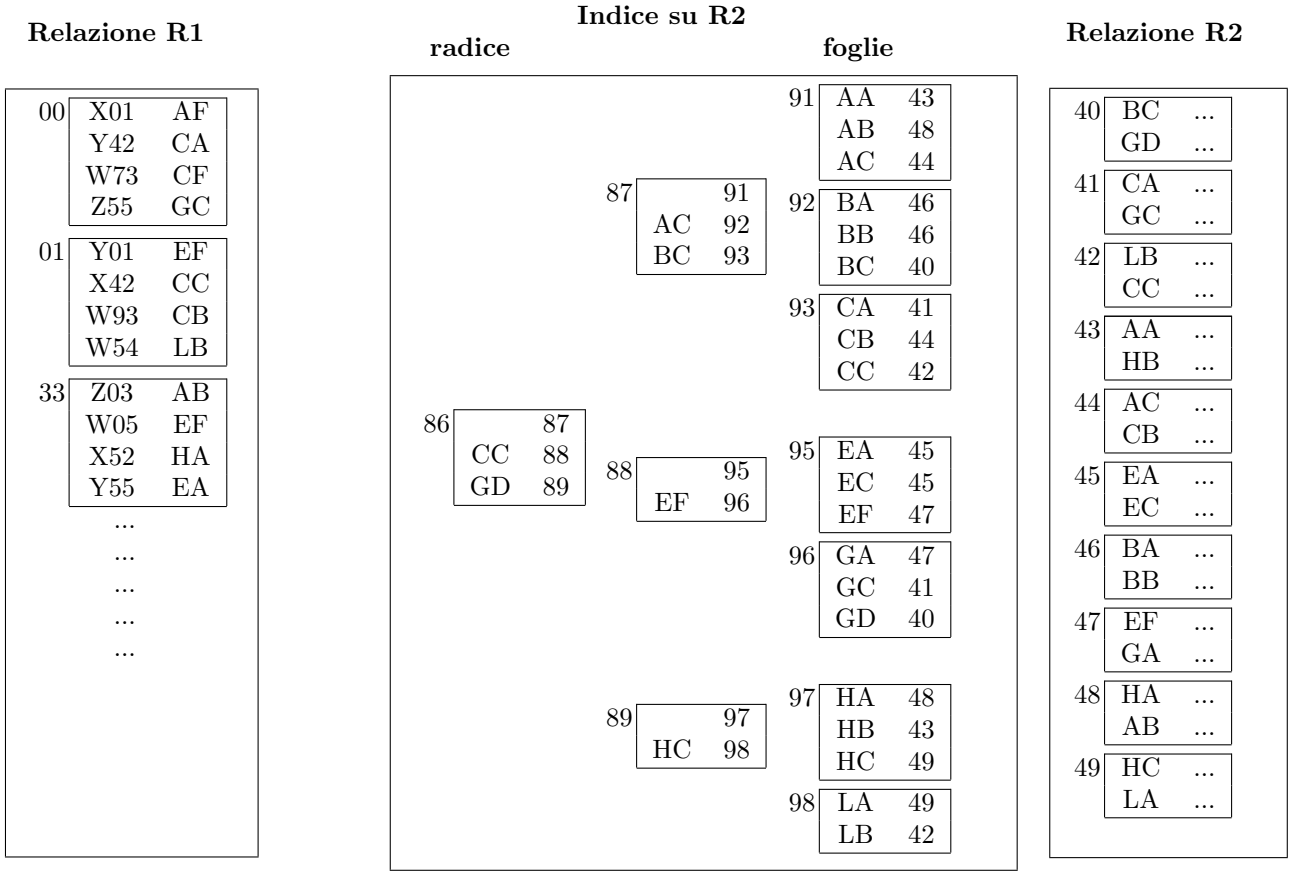
blocchi di R: (indicare nel seguito con B_a)	blocchi di R_1 (indicare nel seguito con B_{b1}) blocchi di R_2 (indicare nel seguito con B_{b2})
costo di o_{1a} (indicare nel seguito con c_{1a})	costo di o_{1b} (indicare nel seguito con c_{1b})
costo di o_{2a} (indicare nel seguito con c_{2a})	costo di o_{2b} (indicare nel seguito con c_{2b})

Le due basi di dati (a) e (b) possono essere considerate due alternative di memorizzazione e le operazioni o_{1a} e o_{1b} sono fra loro equivalenti, così come lo sono o_{2a} e o_{2b}. Quindi è interessante valutare quale delle due alternative sia conveniente.

Supponendo che l'operazione o_1 venga eseguita con frequenza f e l'operazione o_2 con frequenza $k \times f$, si intuisce che (essendo paragonabili, pur se diversi, i costi delle due operazioni) una delle alternative risulta conveniente per k molto maggiore di 1 e l'altra per k molto minore di 1. Verificare questa affermazione con riferimento a $k = 10$ e $k = 0,1$.

Basi di dati II — 30 marzo 2015 — Compito C

Domanda 3 (30%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

Domanda 4 (20%)
Si consideri un B-tree con nodi intermedi che contengono quattro chiavi e cinque puntatori e foglie con quattro chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 40, 54, 14, 32, 22, 23, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Mostrare poi l'albero dopo l'eliminazione della chiave 22 dall'ultimo albero ottenuto in risposta alla domanda precedente.

Basi di dati II — Prova parziale — 30 marzo 2015 — Compito D

Tempo a disposizione: un'ora e quindici minuti.

Si suggerisce di scrivere prima una brutta copia, per indicare poi negli spazi le risposte e brevi giustificazioni.

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (15%)

Considerare una tabella `R` appena creata (e quindi vuota), con le seguenti ipotesi

- `R` è definita su due campi, `A` di lunghezza $a = 6$ byte e `B` di lunghezza $b = 32$ byte, senza vincoli espliciti di chiave (e quindi le operazioni si possono fare senza verifiche particolari);
- la struttura fisica utilizzata per `R` è heap, senza indici, con una memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi ne servano 2 ulteriori per la memorizzazione) e in cui si marcano come liberi gli spazi dei record eliminati, **senza riutilizzarli per successivi inserimenti** (se non dopo una **riorganizzazione** che ricompatti i blocchi);
- il sistema utilizza blocchi di dimensione $D = 4$ Kbyte (approssimabili a 4000).

In tale contesto, supporre che vengano eseguite le seguenti operazioni

1. inserimento di $L = 100.000$ ennuple
2. eliminazione di $L/2 = 50.000$ ennuple (sulla base di una condizione verificabile durante la scansione)
3. dopo la conclusione e la chiusura della scansione precedente, inserimento di altre L ennuple
4. riorganizzazione del file con ricompattazione dei blocchi

Rispondere alle domande seguenti, indicando formule e valori numerici:

Fattore di blocco f per la relazione `R`:
Numero dei blocchi occupati da `R` dopo la prima serie di inserimenti (punto 1):
Numero dei blocchi occupati da `R` dopo le eliminazioni di cui al punto 2:
Numero dei blocchi occupati da `R` dopo la seconda serie di inserimenti (punto 3):
Numero dei blocchi occupati da `R` dopo la ricompattazione (punto 4):

Basi di dati II — 30 marzo 2015 — Compito D

Domanda 2 (30%) Considerare un sistema con blocchi di dimensione $P = 8$ KByte e

(a) una base di dati con una relazione *R(A B C D E)*, in cui gli attributi hanno tutti la stessa dimensione $a = 20$ Byte, . Si supponga che la relazione contenga $L = 40.000.000$ ennuple

(b) una base di dati con una coppia di relazioni R_1*(A B C)* e R_2*(A D E)* ottenute per proiezione dalla relazione R di cui al punto (a)

e le operazioni seguenti:

o_{1a} `SELECT A, B, C FROM R` (su (a))
o_{1b} `SELECT A, B, C FROM R1` (su (b))
o_{2a} `SELECT * FROM R ORDER BY A` (su (a))
o_{2b} `SELECT * FROM R1 JOIN R2 ON R1.A = R2.A ORDER BY R1.A` (su (b))

Indicare i costi relativi all'esecuzione delle varie operazioni, supponendo che, in tutti i casi, l'ordinamento possa essere realizzato con due passate di un merge sort a più vie (e che il join possa essere effettuato con un merge-scan senza materializzare il risultato dei due ordinamenti).

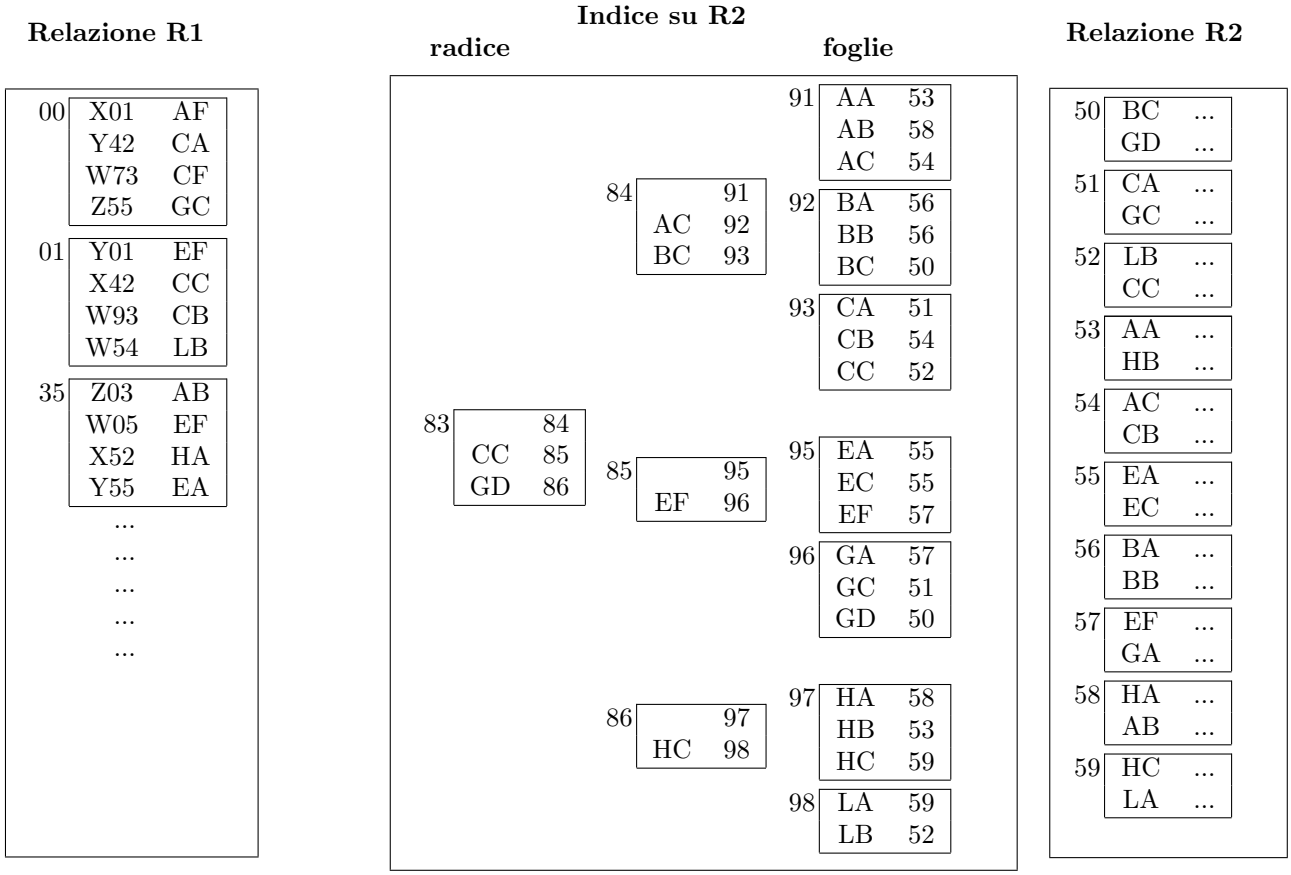
blocchi di R: (indicare nel seguito con B_a)	blocchi di R_1 (indicare nel seguito con B_{b1}) blocchi di R_2 (indicare nel seguito con B_{b2})
costo di o_{1a} (indicare nel seguito con c_{1a})	costo di o_{1b} (indicare nel seguito con c_{1b})
costo di o_{2a} (indicare nel seguito con c_{2a})	costo di o_{2b} (indicare nel seguito con c_{2b})

Le due basi di dati (a) e (b) possono essere considerate due alternative di memorizzazione e le operazioni o_{1a} e o_{1b} sono fra loro equivalenti, così come lo sono o_{2a} e o_{2b}. Quindi è interessante valutare quale delle due alternative sia conveniente.

Supponendo che l'operazione o_1 venga eseguita con frequenza f e l'operazione o_2 con frequenza $k \times f$, si intuisce che (essendo paragonabili, pur se diversi, i costi delle due operazioni) una delle alternative risulta conveniente per k molto maggiore di 1 e l'altra per k molto minore di 1. Verificare questa affermazione con riferimento a $k = 10$ e $k = 0,1$.

Basi di dati II — 30 marzo 2015 — Compito D

Domanda 3 (30%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

Domanda 4 (20%)
Si consideri un B-tree con nodi intermedi che contengono quattro chiavi e cinque puntatori e foglie con quattro chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 42, 53, 17, 32, 21, 22, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Mostrare poi l'albero dopo l'eliminazione della chiave 21 dall'ultimo albero ottenuto in risposta alla domanda precedente.

Basi di dati II — Prova parziale — 30 marzo 2015 — Compito A

Cenni sulle soluzioni
(solo Compito A, le varianti del testo sono in rosso)

Tempo a disposizione: un'ora e quindici minuti.
Si suggerisce di scrivere prima una brutta copia, per indicare poi negli spazi le risposte e brevi giustificazioni.

Cognome _____ **Nome** _____ **Matricola** _____

Domanda 1 (15%)
Considerare una tabella `R` appena creata (e quindi vuota), con le seguenti ipotesi

- `R` è definita su due campi, `A` di lunghezza $a = 6$ byte e `B` di lunghezza $b = 12$ byte, senza vincoli espliciti di chiave (e quindi le operazioni si possono fare senza verifiche particolari);
- la struttura fisica utilizzata per `R` è heap, senza indici, con una memorizzazione a lunghezza fissa (in cui supponiamo che, oltre ai byte necessari per i campi ne servano 2 ulteriori per la memorizzazione) e in cui si marcano come liberi gli spazi dei record eliminati, **senza riutilizzarli per successivi inserimenti** (se non dopo una **riorganizzazione** che ricompatti i blocchi);
- il sistema utilizza blocchi di dimensione $D = 2$ Kbyte (approssimabili a 2000).

In tale contesto, supporre che vengano eseguite le seguenti operazioni

1. inserimento di $N = 100.000$ ennuple
2. eliminazione di $N/2 = 50.000$ ennuple (sulla base di una condizione verificabile durante la scansione)
3. dopo la conclusione e la chiusura della scansione precedente, inserimento di altre N ennuple
4. riorganizzazione del file con ricompattazione dei blocchi

Rispondere alle domande seguenti, indicando formule e valori numerici:

Fattore di blocco f per la relazione `R`: $f = D/(a+b+2) = 2000/20 = 100$
Numero dei blocchi occupati da `R` dopo la prima serie di inserimenti (punto 1): $N/f = 100.000/100 = 1000$
Numero dei blocchi occupati da `R` dopo le eliminazioni di cui al punto 2: $N/f = 100.000/100 = 1000$ (lo spazio libero non viene riutilizzato)
Numero dei blocchi occupati da `R` dopo la seconda serie di inserimenti (punto 3): $2 \times N/f = 2000$ (lo spazio libero non viene riutilizzato)
Numero dei blocchi occupati da `R` dopo la ricompattazione (punto 4): $3/2 \times N/f = 1500$ (lo spazio libero viene riutilizzato)

Basi di dati II — 30 marzo 2015 — Compito A

Domanda 2 (30%) Considerare un sistema con blocchi di dimensione $P = 8$ KByte e

(a) una base di dati con una relazione $R(\underline{A}\ B\ C\ D\ E)$, in cui gli attributi hanno tutti la stessa dimensione $a = 40$ Byte, . Si supponga che la relazione contenga $N = 20.000.000$ ennuple

(b) una base di dati con una coppia di relazioni $R_1(\underline{A}\ B\ C)$ e $R_2(\underline{A}\ D\ E)$ ottenute per proiezione dalla relazione R di cui al punto (a)

e le operazioni seguenti:

o_{1a} `SELECT * FROM R ORDER BY A` (su (a))
o_{1b} `SELECT * FROM R1 JOIN R2 ON R1.A = R2.A ORDER BY R1.A` (su (b))
o_{2a} `SELECT A, B, C FROM R` (su (a))
o_{2b} `SELECT A, B, C FROM R1` (su (b))

Indicare i costi relativi all'esecuzione delle varie operazioni, supponendo che, in tutti i casi, l'ordinamento possa essere realizzato con due passate di un merge sort a più vie (e che il join possa essere effettuato con un merge-scan senza materializzare il risultato dei due ordinamenti).

Le risposte sono per il compito A; per gli altri compiti alcune vanno scambiate

blocchi di R: (indicare nel seguito con B_a) $(N \times 5 \times a)/P =$ $(2 \times 10^7 \times 5 \times 40)/(8 \times 10^3) = 5 \times 10^5$	blocchi di R_1 (indicare nel seguito con B_{b1}) $(N \times 3 \times a)/P = 3 \times 10^5$ blocchi di R_2 (indicare nel seguito con B_{b2}) $(N \times 3 \times a)/P = 3 \times 10^5$
costo di o_{1a} (indicare nel seguito con c_{1a}) $3 \times B_a = 1,5 \times 10^6$	costo di o_{1b} (indicare nel seguito con c_{1b}) $3 \times (B_{b1} + B_{b2}) = 1,8 \times 10^6$
costo di o_{2a} (indicare nel seguito con c_{2a}) $B_a = 0,5 \times 10^6$	costo di o_{2b} (indicare nel seguito con c_{2b}) $B_{b1} = 0,3 \times 10^6$

Le due basi di dati (a) e (b) possono essere considerate due alternative di memorizzazione e le operazioni o_{1a} e o_{1b} sono fra loro equivalenti, così come lo sono o_{2a} e o_{2b}. Quindi è interessante valutare quale delle due alternative sia conveniente.

Supponendo che l'operazione o_1 venga eseguita con frequenza f e l'operazione o_2 con frequenza $k \times f$, si intuisce che (essendo paragonabili, pur se diversi, i costi delle due operazioni) una delle alternative risulta conveniente per k molto maggiore di 1 e l'altra per k molto minore di 1. Verificare questa affermazione con riferimento a $k = 10$ e $k = 0,1$.

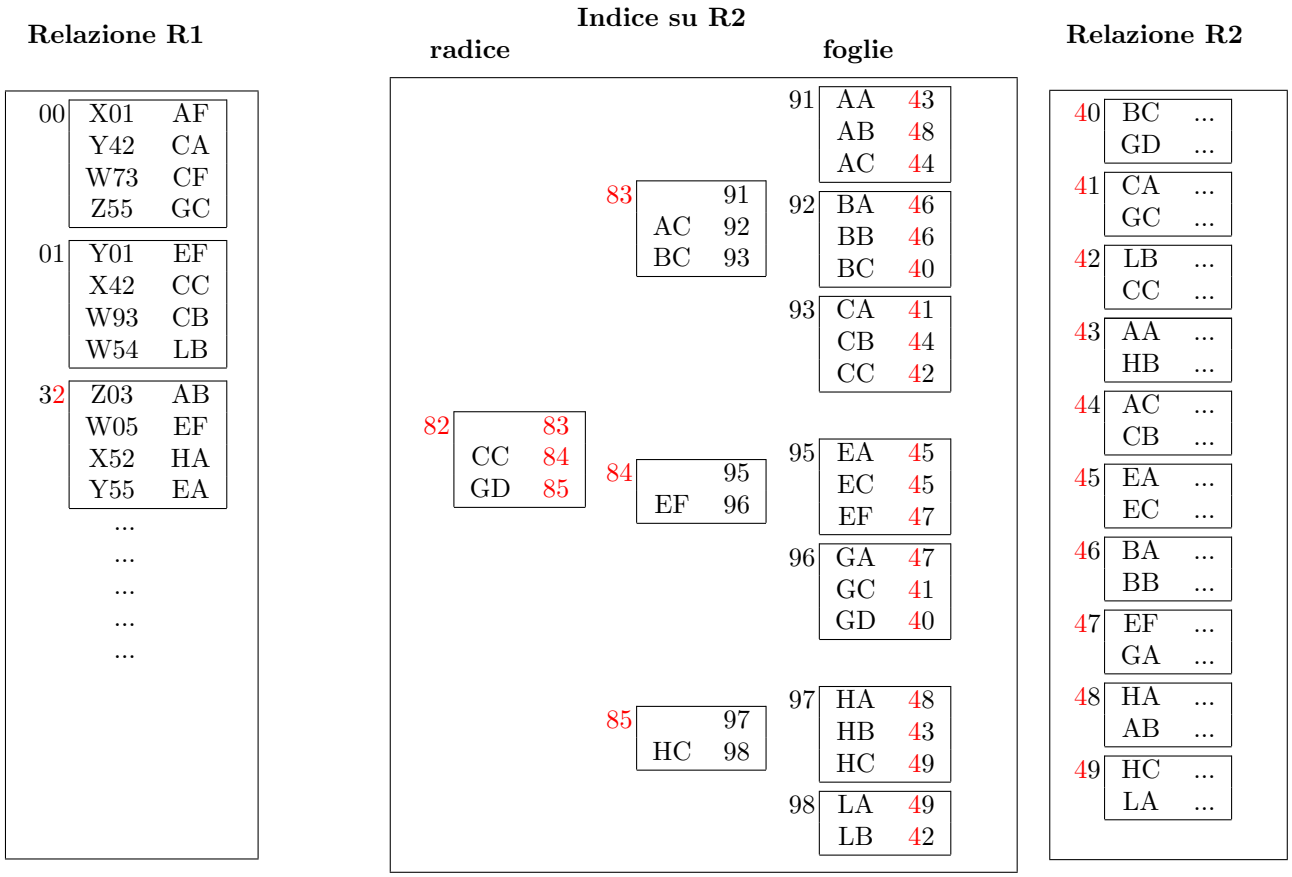
Costo complessivo: $c_1 \times f + c_2 \times k \times f$

Caso (a), k=10: costo complessivo $6,5 \times f \times 10^6$
Caso (b), k=10: costo complessivo $4,8 \times f \times 10^6$
quindi per k grande conviene (b); qui si nota appena, ma al crescere di k si noterebbe di più

Caso (a), k=0,1: costo complessivo $1,55 \times f \times 10^6$
Caso (b), k=0,1: costo complessivo $1,83 \times f \times 10^6$
per k molto piccolo conviene (a)

Basi di dati II — 30 marzo 2015 — Compito A

Domanda 3 (30%) Considerare le relazioni R1 ed R2 e l'indice I2 su R2 schematizzati sotto. I riquadri interni indicano i blocchi e il numero a fianco a ciascun riquadro indica l'indirizzo del blocco. Nell'indice, i valori numerici sono riferimenti ai blocchi (blocchi dell'indice, per la radice e il livello intermedio, e blocchi di R2 per le foglie).



Supponendo di disporre di un buffer di **otto** pagine, considerare l'esecuzione del join di R1 ed R2, sulla base dei valori del secondo attributo di R1 e del primo di R2, con un **nested loop con accesso diretto** tramite l'indice di R2.

Indicare gli indirizzi dei blocchi su cui si eseguono operazioni di pin (o fix) per produrre le prime tre ennuple del risultato.

00, 82, 83, 92,
82, 83, 93, 41,
82, 84, 95,
82, 84, 96, 41,
01, 82, 84, 95, 47

Assumendo una politica di rimpiazzo *LRU*, indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato.

00, 82, 83, 92, 93, 41, 84, 95, 96, 01, 47

In tal caso, indicare gli indirizzi dei blocchi che si può presumere si trovino nei buffer nel momento in cui si produce la terza ennupla.

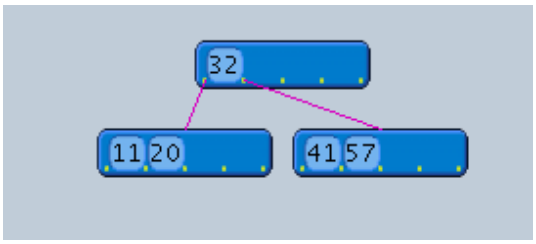
00, 82, 41, 84, 95, 96, 01, 47

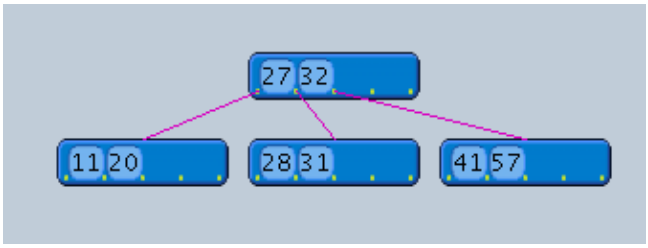
Indicare gli indirizzi dei blocchi effettivamente letti da memoria secondaria e caricati nel buffer (nell'ordine) per produrre le prime tre ennuple del risultato, con riferimento ad una politica di rimpiazzo *clock*.

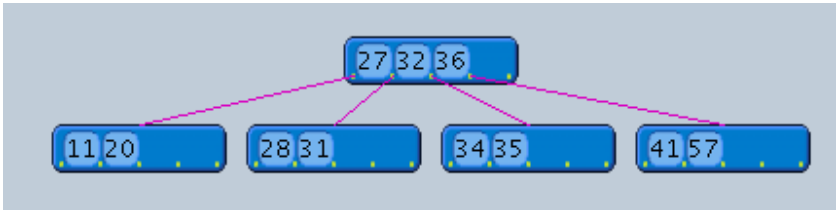
00, 82, 83, 92, 93, 41, 84, 95, 96, 01, 47

Domanda 4 (20%)
Si consideri un B-tree con nodi intermedi che contengono quattro chiavi e cinque puntatori e foglie con quattro chiavi, in cui vengano inserite chiavi (a partire dall'albero vuoto) nel seguente ordine: 41, 57, 11, 32, 20, 27, 28, 31, 34, 35, 36. Mostrare l'albero dopo l'inserimento di cinque chiavi, di otto chiavi e alla fine.

Vengono mostrate le soluzioni per il compito A. Le altre sono analoghe (isomorfe).







Mostrare poi l'albero dopo l'eliminazione della chiave 20 dall'ultimo albero ottenuto in risposta alla domanda precedente.